

AD 662007

INSTRUCTIONS FOR USING EXPERIMENTAL  
0-1 INTEGER LINEAR PROGRAMMING CODE RIP23J

A. M. Geoffrion

A. B. Nelson

November 1967

DNC  
NOV 29 1967  
E

This document has been approved  
for public release and sale; its  
distribution is unlimited.

P-3711

- / -

INSTRUCTIONS FOR USING EXPERIMENTAL  
0-1 INTEGER LINEAR PROGRAMMING CODE RIP23J

A. M. Geoffrion\*  
A. B. Nelson

Consultant to The RAND Corporation, Santa Monica, California

Numerous requests have been received for copies of the experimental code used to obtain the computational experience reported in Refs. 2 and 3. It should be recognized that this is not a production code. It was developed to test the usefulness of certain innovations applied to a simple Balasian algorithm. The central concern was the rate of increase of solution time as a function of the number of variables, rather than how to achieve the smallest possible execution time for particular problems. For this reason, the simplest possible Balasian algorithm was used as the starting point, and concessions were freely made to programming expediency (e.g., no machine language). It would not be difficult to reduce execution times substantially by reprogramming and introducing some of the more sophisticated tests already available in the literature.

We discuss input in Sec. 1; output in Sec. 2; and give an example in Appendix A, and a program listing in Appendix B. For an outline of the working details of the algorithm, see [1] and [2]. Familiarity with these papers is presumed here.

The program solves integer linear programs of the form

(P)                      Minimize  $cx$  subject to  $b + Ax \geq 0$

$$x_j = 0 \text{ or } 1$$

---

\* Any views expressed in this paper are those of the author. They should not be interpreted as reflecting the views of The RAND Corporation or the official opinion or policy of any of its governmental or private research sponsors. Papers are reproduced by The RAND Corporation as a courtesy to members of its staff.

where  $c$  and  $x$  are  $n$ -vectors,  $b$  is an  $m$ -vector, and  $A$  is  $m$  by  $n$ . Any bounded integer linear program can be written in this form, using elementary manipulations if necessary.

## 1. INPUT

The following parameter and data cards appear for each problem to be run:

- (a) Parameter card
- (b) S-card(s)
- (c) C-card(s)
- (d) B-card(s)
- (e) A-card(s)
- (f) Blank card.

Problems can be stacked by repetition of cards a through f.

### Parameter Card

The input parameters are:

- |       |   |
|-------|---|
| M     | The number of constraints   |
| N     | The number of variables   |
| L     | The number of variables in the initial partial solution (L must correspond to the number of entries on the S-card). If $L = 0$ , the initial partial solution is empty. If $L < 0$ , the initial partial solution consists of all variables fixed at the value 0.   |
| SC    | Punch 0 if no imbedded linear program is desired (the algorithm then reduces to a simple Balasian algorithm), and 1 if the imbedded linear program is to be used.   |
| KENUM | When intermediate output is used ( $N\phi P = 0$ ), the fraction of all $2^n$ possible solutions that have been implicitly enumerated is printed out every KENUM times that backtracking occurs. KENUM = 20 is reasonable.  |
| ZBAR  | If an upper bound $\bar{z}$ on the optimal value of the objective function of (P) is known, put $ZBAR = \bar{z} - lcd + 0.0001$ , where lcd is the least common denominator of the cost coefficients $c_j$ (we assume that $\bar{z}$ is a multiple of lcd). Hence, if all $c_j$ (and $\bar{z}$ ) are integer, put $ZBAR = \bar{z} - .9999$ . The effect will be that the program looks only for feasible solutions with value $< ZBAR$ . If no upper bound is known, put $ZBAR = 0$ . See Remark 2 below. |

ISCMAX The maximum number of composite constraints that will be carried. ISCMAX = 4 is reasonable.

ISCFR The frequency with which the imbedded linear program is used. ISCFR = 0 means that it will never be used; ISCFR = j, j a positive integer, means that it will be used every j<sup>th</sup> time. ISCFR = 1 has proven effective, but frequently a value of 8 or so is even better.

MAXC If equal to 0, nothing will happen. If equal to 1, all signs on the C- and A-Cards will be reversed automatically when these cards are read in. This is purely a convenience for manuscripting and key-punching for problems with a preponderance of minus signs in C and A.

MAXT Terminates the calculations after MAXT seconds.

NØP If equal 1, intermediate output will be suppressed; if equal 0, intermediate output will appear. Normally NØP will be set at 1.

ZKBAR Put equal to lcd (see ZBAR) minus 1. Thus, if all  $c_j$  are integer, put ZKBAR = 0. The effect is that the program looks only for feasible solutions with value at least (ZKBAR + .99999) less than the best feasible solution currently known; this doesn't exclude any optimal solutions. (A solution within  $\Delta$  of the optimum can be found if desired by increasing the above value of ZKBAR by  $\Delta$ .)

H1,H2 Arbitrary problem identifiers.

Remark 1: The program is currently dimensioned to use 32,000 words of core in such a way that the following limits must be observed:

$$M + \text{ISCMAX} \leq 50$$

$$N \leq 90.$$

Remark 2: If any  $c_j$  are negative (after MAXC has changed the input signs, if it has value 1), the program internally makes a trivial change of variables to make such  $c_j$  nonnegative: replace  $x_j$  by  $y_j = (1 - x_j)$  if  $c_j < 0$ . The problem is solved in terms of the new variables, and the reverse transformation is made at final output in order to recover the solution to the original problem. ZBAR must be set at a value corresponding to the transformed problem when it is desired to use a

known upper bound; hence, when  $c_j < 0$  for  $j \in J$ , put  $ZBAR = \bar{z} - lcd + .0001 + \sum_{j \in J} |c_j|$ .

The fields and formats of the parameter card are as follows:

<u>Parameter</u>	<u>Column</u>	<u>Format</u>
M	1-3	Integer
N	4-6	"
L	7-9	"
SC	10-12	"
KENUM	13-17	"
ZBAR	18-23	E
ISCMAX	24-26	Integer
ISCFR	27-29	"
MAXC	30-32	"
MAXT	33-37	"
NØP	38-40	"
ZKBAR	41-46	E
H1	47-52	Hollerith
H2	53-58	"

#### S-Card(s)

The algorithm can start with any initial partial solution (see [1]). When the initial partial solution is desired to be nonempty ( $L > 0$ ), if  $x_j$  is to be fixed at the value one (zero) then "j" ("-j") is entered on the S-card, followed by "B" when an underline is desired. The S-card is divided into 12 fields of 5 columns each: 1-5, 6-10, ..., 66-70. Only the first four columns of each field are to be used except when underlines are desired, in which case "B" must appear in the fifth column of the field.

The special instruction given above in Remark 2 for ZBAR, when a change of variables is made, also applies here. That is, the sign of  $\pm j$  or  $\pm jB$  must be changed when  $c_j < 0$ .

#### C-Card(s)

The values of the  $c_j$  must be entered in order (negative values are permissible, as noted above). Each card has six fields of eleven columns read in E-format. The fields are separated by an unread column so that the values of the  $c_j$  are in columns 1-11, 13-23, ..., 61-71.

B-Card(s)

The values of the  $b_i$  must be entered in order. The format is exactly the same as for the C-cards.

A-Card(s)

Only nonzero  $a_{ij}$  need be entered, and they may be entered in any order. Each value is identified by its row and column. There are four or fewer entries on each of the "A" cards. Each entry has a seventeen column field.

	<u>Columns</u>	<u>Format</u>
Row	1-3	Integer
Column	4-6	Integer
Value	7-17	E

The fields are separated by an unread column so that the matrix subscripts and values of the  $a_{ij}$  are in columns 1-17, 19-35, 37-53, 55-71.

## 2. OUTPUT

The preliminary, intermediate, and final outputs are as follows.

The parameter, "S", "C", and "B" cards are printed in that order (six values to a line for the "C" and "B" cards). Then the complete A matrix is printed (with zeros), row by row. If MAXC = 1, the sign reversals in "C" and "A" will be seen to have occurred. If a change of variables was made internally, the new c, b, and A are printed out (if no change of variables was necessary, the identical c, b, and A are printed out again anyway).

If NØP = 0, intermediate output is produced to reveal the course of the calculations - each feasible solution found, each new composite constraint, data concerning each imbedded linear program, and a summary of progress to date after each KENUM "backtrackings." Since this information is likely to be of little incremental value to the user over the final output information, no detailed explanation is given here.

The final output gives the problem designation; the message "implicit enumeration complete" or "time exceeded" according as termination did or did not occur within MAXT seconds; the total execution time in seconds; the solution (obj. fc. value and a list of which variables equal 1) both before and after the variable change (if no variable change occurred, these solutions are identical); and some statistical information on the course of the algorithm, such as the number of feasible solutions found, the number of times the imbedded linear program was solved, the number of iterations, and the time at which the last feasible solution was found. In the event that no feasible solutions were found, this is indicated by the zeros in the solution after variable change and the statistic "no. feasible solutions, 0." In the event that the time limit was exceeded, the final output is preceded by a brief report giving the proportion of all  $2^n$  possible solutions that have been accounted for and the final "state" vector [1], with "B" signifying an underline. All the information needed to restart the calculations is available: make the S-card correspond to the final state vector (set L accordingly), and put ZBAR equal to LEAST Z AFTER VARIABLE CHANGE - lcd + 0.0001.

Appendix A

EXAMPLE

We shall illustrate the above by solving Petersen's fifth example [4].

For this problem,  $M = 10$  and  $N = 28$ . We shall take  $L = 28$ ,  $SC = 1$ ,  $KENUM = 20$ ,  $ZBAR = 0$  (since we will not bother to determine a bound on the objective function),  $ISCMAX = 4$ ,  $ISCFR = 1$ ,  $MAXC = 1$  (since we wish to avoid keypunching all the minus signs for  $c$  and  $A$ ),  $MAXT = 60$ ,  $NOP = 1$ ,  $ZKBAR = 4$  (since the least common denominator of the  $c_j$  is 5), and  $H1 = PETE 5$ .

The S-card will contain the numbers 1, 2, ..., 28 (we have elected an initial partial solution with all variables fixed at the value 1).

The output is reproduced below.

10 28 28 1 20 0. 4 1 1 60 1 0.4000E 01 PETE 5

**M = 10      N = 28**

1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28
-1.0000000	E 02	-2.2000000	E 02	-9.0000000	E 01	-4.0000000	E 02	-3.0000000	E 02	-4.0000000	E 02		
-2.0500000	E 02	-1.2000000	E 02	-1.6000000	E 02	-5.8000000	E 02	-4.0000000	E 02	-1.4000000	E 02		
-1.0000000	E 02	-1.3000000	E 03	-6.5000000	E 02	-3.2000000	E 02	-4.8000000	E 02	-8.0000000	E 01		
-6.0000000	E 01	-2.5500000	E 03	-3.1000000	E 03	-1.1000000	E 03	-9.5000000	E 02	-4.5000000	E 02		
-3.0000000	E 02	-2.2000000	E 02	-2.0000000	E 02	-5.2000000	E 02						
9.3000000	E 02	1.2100000	E 03	2.7200000	E 02	4.6200000	E 02	5.3200000	E 02	5.7200000	E 02		
2.4000000	E 02	4.0000000	E 02	4.7000000	E 02	4.9000000	E 02						
-8.0000000	E 00	-2.4000000	E 01	-1.3000000	E 01	-8.0000000	E 01	-7.0000000	E 01	-8.0000000	E 01		
-4.5000000	E 01	-1.5000000	E 01	-2.8000000	E 01	-9.0000000	E 01	-1.3000000	E 02	-3.2000000	E 01		
-2.0000000	E 01	-1.2000000	E 02	-4.0000000	E 01	-3.0000000	E 01	-2.0000000	E 01	-6.0000000	E 00		
-3.0000000	E 00	-1.8000000	E 02	-2.2000000	E 02	-5.0000000	E 01	-3.0000000	E 01	-5.0000000	E 01		
-1.2000000	E 01	-5.0000000	E 00	-8.0000000	E 00	-1.8000000	E 01						
-8.0000000	E 00	-4.4000000	E 01	-1.3000000	E 01	-1.0000000	E 02	-1.0000000	E 02	-9.0000000	E 01		
-7.5000000	E 01	-2.5000000	E 01	-2.8000000	E 01	-1.2000000	E 02	-1.3000000	E 02	-3.2000000	E 01		
-4.0000000	E 01	-1.6000000	E 02	-4.0000000	E 01	-6.0000000	E 01	-5.5000000	E 01	-1.0000000	E 01		
-6.0000000	E 00	-2.4000000	E 02	-2.9000000	E 02	-8.0000000	E 01	-9.0000000	E 01	-7.0000000	E 01		
-2.7000000	E 01	-1.7000000	E 01	-8.0000000	E 00	-2.8000000	E 01						
-3.0000000	E 00	-6.0000000	E 00	-4.0000000	E 00	-2.0000000	E 01	-2.0000000	E 01	-3.0000000	E 01		
-8.0000000	E 00	-3.0000000	E 00	-1.2000000	E 01	-1.4000000	E 01	-4.0000000	E 01	-6.0000000	E 00		
-3.0000000	E 00	-2.0000000	E 01	-5.0000000	E 00	-0.		-5.0000000	E 00	-3.0000000	E 00		
-0.		-2.0000000	E 01	-3.0000000	E 01	-4.0000000	E 01	-1.0000000	E 01	-0.			
-5.0000000	E 00	-0.		-0.		-1.0000000	E 01						
-5.0000000	E 00	-9.0000000	E 00	-6.0000000	E 00	-4.0000000	E 01	-3.0000000	E 01	-4.0000000	E 01		
-1.6000000	E 01	-5.0000000	E 00	-1.8000000	E 01	-2.4000000	E 01	-6.0000000	E 01	-1.6000000	E 01		
-1.1000000	E 01	-3.0000000	E 01	-2.5000000	E 01	-1.0000000	E 01	-1.3000000	E 01	-5.0000000	E 00		
-1.0000000	E 00	-8.0000000	E 01	-6.0000000	E 01	-5.0000000	E 01	-2.0000000	E 01	-3.0000			

-3.0000000 E 00	-4.0000000 F 00	-5.0000000 E 00	-2.0000000 E 01	-1.4000000 E 01	-2.0000000 E 01
-6.0000000 F 00	-1.2000000 E 01	-1.0000000 E 01	-1.8000000 E 01	-4.2000000 E 01	-9.0000000 E 00
-1.2000000 E 01	-1.0000000 F 02	-2.0000000 E 01	-5.0000000 E 00	-6.0000000 E 00	-4.0000000 E 00
-1.0000000 E 00	-2.0000000 E 01	-5.0000000 F 01	-3.0000000 E 01	-5.0000000 E 00	-2.0000000 E 01
-2.0000000 E 01	-1.0000000 E 01	-1.0000000 E 01	-2.0000000 E 01		
-3.0000000 E 00	-6.0000000 E 00	-9.0000000 E 00	-3.0000000 E 01	-2.9000000 E 01	-2.0000000 E 01
-1.2000000 E 01	-1.2000000 E 01	-1.0000000 E 01	-3.0000000 E 01	-4.2000000 E 01	-1.8000000 F 01
-1.8000000 F 01	-1.1000000 E 02	-2.0000000 E 01	-1.5000000 E 01	-1.8000000 E 01	-7.0000000 E 00
-2.0000000 F 00	-4.0000000 F 01	-6.0000000 E 01	-5.0000000 E 01	-2.5000000 E 01	-2.5000000 F 01
-2.5000000 E 01	-1.5000000 E 01	-1.0000000 E 01	-2.8000000 E 01		
-3.0000000 E 00	-8.0000000 E 00	-9.0000000 E 00	-3.5000000 E 01	-2.9000000 E 01	-2.0000000 E 01
-1.6000000 E 01	-1.5000000 E 01	-1.0000000 E 01	-3.0000000 E 01	-4.2000000 E 01	-2.0000000 E 01
-1.8000000 F 01	-1.2000000 E 02	-2.0000000 E 01	-2.0000000 F 01	-2.2000000 E 01	-7.0000000 E 00
-3.0000000 E 00	-5.0000000 E 01	-6.0000000 E 01	-5.5000000 E 01	-2.5000000 E 01	-3.0000000 F 01
-2.5000000 E 01	-1.5000000 E 01	-1.0000000 E 01	-2.8000000 E 01		
1.0000000 E 02	2.2000000 E 02	9.0000000 E 01	4.0000000 E 02	3.0000000 E 02	4.0000000 E 02
2.0500000 E 02	1.2000000 E 02	1.6000000 E 02	5.8000000 E 02	4.0000000 E 02	1.4000000 E 02
1.0000000 E 02	1.3000000 F 03	6.5000000 E 02	3.2000000 E 02	4.8000000 E 02	8.0000000 E 01
6.0000000 E 01	2.5500000 E 03	3.1000000 E 03	1.1000000 E 03	9.5000000 E 02	4.5000000 E 02
3.0000000 E 02	2.2000000 E 02	2.0000000 E 02	5.2000000 E 02		
-4.9700000 E 02	-7.7600000 E 02	-4.5000000 E 01	-1.8000000 E 02	-2.5400000 E 02	-2.4900000 E 02
1.3000000 E 01	-9.6000000 E 01	-2.1900000 E 02	-2.5500000 E 02		
8.0000000 F 00	2.4000000 E 01	1.3000000 E 01	8.0000000 E 01	7.0000000 E 01	8.0000000 E 01
4.5000000 E 01	1.5000000 E 01	2.8000000 E 01	9.0000000 E 01	1.3000000 E 02	3.2000000 E 01
2.0000000 E 01	1.2000000 E 02	4.0000000 E 01	3.0000000 E 01	2.0000000 E 01	6.0000000 E 00
3.0000000 E 00	1.8000000 E 02	2.2000000 E 02	5.0000000 E 01	3.0000000 E 01	5.0000000 E 01
1.2000000 E 01	5.0000000 E 00	8.0000000 E 00	1.8000000 E 01		
8.0000000 E 00	4.4000000 E 01	1.3000000 E 01	1.0000000 E 02	1.0000000 E 02	9.0000000 E 01
7.5000000 E 01	2.5000000 E 01	2.8000000 E 01	1.2000000 E 02	1.3000000 E 02	3.2000000 E 01
4.0000000 E 01	1.6000000 E 02	4.0000000 E 01	6.0000000 E 01	5.5000000 E 01	1.0000000 E 01
6.0000000 E 00	2.4000000 E 02	2.9000000 E 02	8.0000000 E 01	9.0000000 E 01	7.0000000 E 01
2.7000000 E 01	1.7000000 E 01	8.0000000 E 00	2.8000000 E 01		
3.0000000 F 00	6.0000000 E 00	4.0000000 E 00	2.0000000 E 01	2.0000000 E 01	3.0000000 E 01
8.0000000 E 00	3.0000000 E 00	1.2000000 E 01	1.4000000 F 01	4.0000000 E 01	6.0000000 F 00
3.0000000 E 00	2.0000000 E 01	5.0000000 E 00	0.	5.0000000 E 00	3.0000000 E 00
0.	2.0000000 E 01	3.0000000 E 01	4.0000000 E 01	1.0000000 E 01	0.
5.0000000 F 00	0.	0.	1.0000000 E 01		
5.0000000 E 00	9.0000000 E 00	6.0000000 E 00	4.0000000 E 01	3.0000000 E 01	4.0000000 E 01
1.6000000 E 01	5.0000000 E 00	1.8000000 E 01	2.4000000 E 01	6.0000000 E 01	1.6000000 F 01
1.1000000 E 01	3.0000000 E 01	2.5000000 E 01	1.0000000 E 01	1.3000000 E 01	5.0000000 E 00
1.0000000 E 00	8.0000000 E 01	6.0000000 E 01	5.0000000 E 01	2.0000000 E 01	3.0000000 E 01
1.0000000 E 01	5.0000000 E 00	3.0000000 E 00	2.0000000 E 01		
5.0000000 F 00	1.1000000 E 01	7.0000000 E 00	5.0000000 E 01	4.0000000 E 01	4.0000000 E 01
1.9000000 E 01	7.0000000 E 00	1.8000000 E 01	2.9000000 E 01	7.0000000 E 01	2.1000000 E 01
1.7000000 E 01	3.0000000 E 01	2.5000000 E 01	1.5000000 E 01	2.5000000 E 01	5.0000000 E 00
1.0000000 E 00	1.0000000 E 02	7.0000000 E 01	5.5000000 E 01	2.0000000 F 01	5.0000000 E 01
1.5000000 E 01	1.5000000 E 01	6.0000000 E 00	2.0000000 E 01		
5.0000000 E 00	1.1000000 E 01	7.0000000 E 00	5.5000000 E 01	4.0000000 E 01	4.0000000 E 01
2.1000000 F 01	9.0000000 E 00	1.8000000 E 01	2.9000000 E 01	7.0000000 E 01	2.1000000 E 01
1.7000000 E 01	3.5000000 E 01	2.5000000 E 01	2.0000000 E 01	2.5000000 E 01	5.0000000 E 00
7.0000000 E 00	1.1000000 E 02	7.0000000 E 01	5.5000000 E 01	2.0000000 E 01	5.0000000 E 01

2.0000000 F 01	1.5000000 F 01	6.0000000 F 00	2.0000000 F 01		
0.	0.	1.0000000 F 00	1.0000000 F 01	4.0000000 F 00	1.0000000 F 01
0.	6.0000000 F 00	0.	6.0000000 F 00	3.2000000 F 01	3.0000000 F 00
0.	7.0000000 F 01	1.0000000 F 01	0.	0.	0.
0.	0.	3.0000000 F 01	1.0000000 F 01	0.	1.0000000 F 01
1.0000000 F 01	5.0000000 F 00	0.	1.0000000 F 01		
3.0000000 F 00	4.0000000 F 00	5.0000000 F 00	2.0000000 F 01	1.4000000 F 01	2.0000000 F 01
6.0000000 F 00	1.2000000 F 01	1.0000000 F 01	1.8000000 F 01	4.2000000 F 01	9.0000000 F 00
1.2000000 F 01	1.0000000 F 02	2.0000000 F 01	5.0000000 F 00	6.0000000 F 00	4.0000000 F 00
1.0000000 F 00	2.0000000 F 01	5.0000000 F 01	3.0000000 F 01	5.0000000 F 00	2.0000000 F 01
2.0000000 F 01	1.0000000 F 01	1.0000000 F 01	2.0000000 F 01		
3.0000000 F 00	6.0000000 F 00	9.0000000 F 00	3.0000000 F 01	2.9000000 F 01	2.0000000 F 01
1.2000000 F 01	1.2000000 F 01	1.0000000 F 01	3.0000000 F 01	4.2000000 F 01	1.8000000 F 01
1.8000000 F 01	1.1000000 F 02	2.0000000 F 01	1.5000000 F 01	1.8000000 F 01	7.0000000 F 00
2.0000000 F 00	4.0000000 F 01	6.0000000 F 01	5.0000000 F 01	2.5000000 F 01	2.5000000 F 01
2.5000000 F 01	1.5000000 F 01	1.0000000 F 01	2.8000000 F 01		
3.0000000 F 00	8.0000000 F 00	9.0000000 F 00	3.5000000 F 01	2.9000000 F 01	2.0000000 F 01
1.6000000 F 01	1.5000000 F 01	1.0000000 F 01	3.0000000 F 01	4.2000000 F 01	2.0000000 F 01
1.8000000 F 01	1.2000000 F 02	2.0000000 F 01	2.0000000 F 01	2.2000000 F 01	7.0000000 F 00
3.0000000 F 00	5.0000000 F 01	6.0000000 F 01	5.5000000 F 01	2.5000000 F 01	3.0000000 F 01
2.5000000 F 01	1.5000000 F 01	1.0000000 F 01	2.8000000 F 01		

PETE 5

IMPLICIT ENUMERATION COMPLETE TOTAL TIME= 4.371

LEAST Z AFTER VARIABLE CHANGE = 3.0950000 E 03

0	0	0	4	5	6	7	8	0	10	11	12	13	0	0
0	0	0	0	0	0	0	0	24	0	0	0	0		

LEAST Z BEFORE VARIABLE CHANGE = -1.2400000 E 04

1	2	3	0	0	0	0	0	9	0	0	0	0	14	15
16	17	18	19	20	21	22	23	0	25	26	27	28		

NO. FEASIBLE SOLUTIONS 24

ZS GE ZBAR 5 TIMES

CONSTRAINT INFEASIBLE 15 TIMES

AUGMENTATION IMPOSSIBLE 2 TIMES

AUGMENTATION POSSIBLE 19 TIMES

INTEGER DUALS 0 TIMES

LP FATHOMED 2 TIMES

LP CALLED 21 TIMES

NO. ITERATIONS 95

LAST FEASIBLE SOLUTION AT 4.132 SECONDS

Appendix B

LISTING OF RIP23J

```

$IBFTC RIP23J                                00000010
        DIMENSION A(50,90),JF(50,90)          00000020
        DIMENSION B(100),C(100),BS(100),S(100),SB(100),NS(100),NF(100) 00000030
        DIMENSION ITEMP(4),JTEMP(4),ATEMP(4),SMAX(100),SMAXB(100),T(100) 00000040
        DIMENSION CS(100),HI(100)             00000050
        DIMENSION XL(90),DI(90),E(90,90)       00000060
        DIMENSION JHI(100),XX(100),Y(100),PE(100),KO(6) 00000070
        INTEGER S,SMAX,SC,T                    00000080
        COMMON /BLS/MS(90),ZBAR                00000090
        DATA BCIB/6HB /                      00000100
        DATA BLANK/6H /                      00000110
100 DO 110 I=1,90                             00000120
        H(I)=0.0                              00000130
        B(I)=0.0                              00000140
        C(I)=0.0                              00000150
        BS(I)=0.0                             00000160
        S(I)=0                                00000170
        SB(I)=BLANK                           00000180
        NS(I)=0                               00000190
        NF(I)=0                               00000200
        SMAX(I)=0                             00000210
        SMAXB(I)=BLANK                       00000220
        T(I)=0                               00000230
        DO 110 J=1,50                         00000240
        A(J,I)=0.0                           00000250
        JF(J,I)=0.0                          00000260
110 CONTINUE                                00000270
        II=0                                 00000280
        NCON=0                               00000290
        NRED=0                               00000300
        NAUG=0                               00000310
        NOPT=0                               00000320
        NID=0                                00000330
        NAP=0                                00000340
        NLPF=0                               00000350
        NSIMP=0                              00000360
        NFATH=0                              00000370
        NENUM=0                              00000380
        NTCE=0                               00000390
        ITB=0                                00000400
        IPOST=1                              00000410
        IINS=5                               00000420
C                                             00000430
C READ A NEW SET OF DATA                    00000440
C PARAMETER CARD FIRST                      00000450
C 'S' CARD THIRD                            00000460
C 'C','B','A' MATRICES FOLLOW 'S'          00000470
C                                             00000480
C MINIMIZE SUM C(I)*X(I)                    00000490
C CONSTRAINTS ARE B(I)+SUM A(I,J)*X(J) GE ZERO 00000500
        READ 9000,M,N,L,SC,KENUM,ZBAR,ISCMAX,ISCFR,MAXC,MAXT, 00000510
        * NOP,ZKBAR,H1,H2                    00000520
9000 FORMAT (4I3,15,E6.0,3I3,15,I3,E6.0,2A6) 00000530
        PRINT 9993                            00000540
        PRINT 9001,M,N,L,SC,KENUM,ZBAR,ISCMAX,ISCFR,MAXC,MAXT, 00000550
        * NOP,ZKBAR,H1,H2                    00000560
9001 FORMAT (4I3,15,1X,E11.4,3I3,15,I3,E11.4,1X,2A6) 00000570
        IF (IINS.EQ.0) IINS=9999             00000580
        IF (MAXT.EQ.0) MAXT=999999           00000590
        MAXT=1000*MAXT                       00000600

```

MO=M	00000610
M1=M0+1	0000062C
JSCFR=JSCFR	00000630
ZKBAR=ZKBAR+.99999	00000640
PRINT 9010,M,N	00000650
9010 FORMAT (3HOM=,I3,2X,2HN=,I3)	00000660
PRINT 9992	00000670
9991 FORMAT (1H )	00000680
9992 FORMAT (1H0)	00000690
9993 FORMAT (1H1)	00000700
L1=L	00000710
IF (L.LE.0) L1=0	00000720
READ 9100,((S(K),SB(K)),K=1,L1)	00000730
9100 FORMAT (14(I4,A1))	00000740
IF (L.GE.0) GO TO 130	00000750
L=N	00000760
DO 120 K=1,N	00000770
120 S(K)=-K	00000780
130 CONTINUE	00000790
READ 9200,(C(J),J=1,N)	00000800
9200 FORMAT (6(E11.0,1X))	00000810
C*****	00000820
IF (MAXC.EQ.0) GO TO 141	00000830
DO 140 J=1,N	00000840
140 C(J)=-C(J)	00000850
141 CONTINUE	00000860
READ 9200,(B(I),I=1,M)	00000870
200 READ 9400,((ITEMP(K),JTEMP(K),ATEMP(K)),K=1,4)	00000880
9400 FORMAT (4(2I3,E11.0,1X))	00000890
END=0.0	00000900
DO 250 K=1,4	00000910
KI=ITEMP(K)	00000920
KJ=JTEMP(K)	00000930
IF (KI.EQ.0) GO TO 250	00000940
IF (KJ.EQ.0) GO TO 250	00000950
KJF=NF(KI)+1	00000960
NF(KI)=KJF	00000970
JF(KI,KJF)=KJ	00000980
C*****	00000990
IF (MAXC.NE.0) ATEMP(K)=-ATEMP(K)	00001000
A(KI,KJ)=ATEMP(K)	00001010
END=1.0	00001020
250 CONTINUE	00001030
IF (END.NE.0.0) GO TO 200	00001040
PRINT 9992	00001050
PRINT 9500,((S(K),SB(K)),K=1,L)	00001060
9500 FORMAT (14(3X,I4,A1))	00001070
PRINT 9992	00001080
PRINT 9600,(C(J),J=1,N)	00001090
PRINT 9992	00001100
PRINT 9600,(B(I),I=1,M)	00001110
PRINT 9992	00001120
DO 251 I=1,M	00001130
PRINT 9600,(A(I,J),J=1,N)	00001140
PRINT 9991	00001150
251 CONTINUE	00001160
PRINT 9992	00001170
C	00001180
C ALL DATA READ FOR THIS RUN	00001190
C	00001200

DO 255 J=1,N	00001210
CS(J)=C(J)	00001220
IF (C(J).GE.0.0) GO TO 255	00001230
C(J)=-C(J)	00001240
DO 253 I=1,M	00001250
B(I)=B(I)+A(I,J)	00001260
253 A(I,J)=-A(I,J)	00001270
255 CONTINUE	00001280
PRINT 9600,(C(J),J=1,N)	00001290
PRINT 9992	00001300
PRINT 9600,(B(I),I=1,M)	00001310
PRINT 9992	00001320
DO 260 I=1,M	00001330
PRINT 9600,(A(I,J),J=1,N)	00001340
PRINT 9991	00001350
260 CONTINUE	00001360
9600 FORMAT (6(2X,1PE15.8))	00001370
IF (ZBAR.GT.0.0) GO TO 300	00001380
ZBAR=0.0	00001390
DO 275 J=1,N	00001400
275 ZBAR=ZBAR+C(J)	00001410
300 ZS=0.0	00001420
DO 325 I=1,M	00001430
325 BS(I)=B(I)	00001440
DO 330 J=1,N	00001450
330 NS(J)=J	00001460
IF (L.EQ.0) GO TO 400	00001470
DO 375 K=1,L	00001480
J1=S(K)	00001490
K1=IABS(J1)	00001500
NS(K1)=0	00001510
IF (J1.LE.0) GO TO 375	00001520
ZS=ZS+C(J1)	00001530
DO 350 I=1,M	00001540
350 BS(I)=BS(I)+A(I,J1)	00001550
375 CONTINUE	00001560
400 CONTINUE	00001570
IF (MO+ISCMAX.GT.50) ISCMAX=50-MO	00001580
I1=MO+ISCMAX	00001590
DO 425 I=M1,I1	00001600
NF(I)=N	00001610
DO 425 J=1,N	00001620
JF(I,J)=J	00001630
425 CONTINUE	00001640
CALL DATIME (0,ITO)	00001650
IT1=ITO	00001660
GO TO 1910	00001670
C	00001680
C INITIALIZATION COMPLETE	00001690
C	00001700
1000 CONTINUE	00001710
IF (SC.EQ.0) GO TO 2400	00001720
C SURROGATE CONSTRAINTS GO HERE	00001730
JSCFR=JSCFR+1	00001740
IF (ISCFR.GT.JSCFR) GO TO 2400	00001750
ML=N-L	00001760
IF (ML.LE.1) GO TO 2400	00001770
JSCFR=0	00001780
1050 DO 1060 J=1,N	00001790
1060 MS(J)=0	00001800

NSIMP=NSIMP+1	00001810
IF(L.EQ.0) GO TO 1076	00001820
DO 1075 I=1,L	00001830
J=IABS(S(I))	00001840
1075 MS(J)=-S(I)	00001850
IF (NOP.NE.0) GO TO 1076	00001860
PRINT 3600,((S(K),SB(K),	00001870
1076 CALL SIMPLE (II, N,MO,A,C, D,XL,D,JH,XX,Y,OBJ,E,NOP)	00001880
IF (NOP.NE.0) GO TO 1077	00001890
PRINT 9600,OBJ,ZBAR	00001900
1077 CONTINUE	00001910
II=II+IPOST	00001920
IF (KO(1).EQ.2) GO TO 3400	00001930
IF (KO(1).EQ.4) GO TO 100	00001940
IF (KO(1).EQ.6) GO TO 1500	00001950
VLPS=-OBJ	00001960
IF (VLPS.LE. (-ZBAR))GO TO 1499	00001970
DO 1350 I=1,N	00001980
IF (D(I).NE.AINT(D(I)).AND.NS(I).NE.0) GO TO 1500	00001990
1350 CONTINUE	00002000
DO 1450 J=1,N	00002010
IF (NS(J).EQ.0) GO TO 1450	00002020
I=J	00002030
L=L+1	00002040
NS(J)=0	00002050
SB(L)=BCIB	00002060
IF (D(I).NE.0.0) GO TO 1400	00002070
S(L)=-J	00002080
GO TO 1450	00002090
1400 S(L)=J	00002100
ZS=ZS+C(J)	00002110
DO 1425 II=1,M	00002120
1425 BS(II)=BS(II)+A(II,J)	00002130
1450 CONTINUE	00002140
NID=NID+1	00002150
GO TO 2320	00002160
1499 KO(1)=6	00002170
1500 IF (ISCMAX.LE.0) GO TO 1599	00002180
BMP1=ZBAR	00002190
DO 1505 I=1,MO	00002200
1505 BMP1=BMP1+XL(I)*B(I)	00002210
IF (ABS(BMP1-B(M)).LE.0.0005) GO TO 1599	00002220
IF (M-MO.LT.ISCMAX) GO TO 1520	00002230
DO 1510 I=M1,M	00002240
B(I)=B(I+1)	00002250
BS(I)=BS(I+1)	00002260
DO 1510 J=1,N	00002270
1510 A(I,J)=A(I+1,J)	00002280
M=M-1	00002290
1520 B(M+1)=BMP1	00002300
DO 1550 J=1,N	00002310
ZJH=XX(J)	00002320
IF (JH(J).GE.(-M)) ZJH=-ZJH	00002330
IF (JH(J).GT.0) ZJH=0.	00002340
1550 A(M+1,J)=ZJH	00002350
M=M+1	00002360
BS(M)=B(M)	00002370
DO 1575 K=1,L	00002380
K1=S(K)	00002390
IF (K1.LE.0) GO TO 1575	00002400

BS(M)=BS(M)+A(M,K1)	00002410
1575 CONTINUE	00002420
IF (NOP.NE.0) GO TO 1599	00002430
PRINT 1598,M	00002440
PRINT 9600,(A(M,J),J=1,N),B(M),BS(M)	00002450
1598 FORMAT (22HOSURROGATE CONSTRAINTS,2X,14)	00002460
1599 IF (KD(1).EQ.6) GO TO 3400	00002470
1900 GO TO 2400	00002480
1910 IJK=0	00002490
1920 CONTINUE	00002500
IF (ZS.GE.ZBAR) GO TO 3100	00002510
DO 1950 I1=1,M0	00002520
1950 IF (BS(I1).LT.0.0) GO TO 1980	00002530
GO TO 2320	00002540
1980 CONTINUE	00002550
DO 2000 J=1,N	00002560
IF (NS(J).EQ.0) GO TO 2000	00002570
IF (ZS+C(J).LT.ZBAR) GO TO 2000	00002580
NS(J)=0	00002590
L=L+1	00002600
SB(L)=BCIB	00002610
S(L)=-J	00002620
2000 CONTINUE	00002630
KINS=0	00002640
IF (IJK.EQ.1) GO TO 2220	00002650
IF (IJK.EQ.2) GO TO 1000	00002660
IJK=1	00002670
IF (M.LT.M1) GO TO 2025	00002680
MSC=0	00002690
I1=M1	00002700
I2=M	00002710
GO TO 2050	00002720
2025 MSC=1	00002730
I1=1	00002740
I2=M0	00002750
2050 DO 2220 I=I1,I2	00002760
C=BS(I)	00002770
DO 2100 J=1,N	00002780
IF (NS(J).EQ.0) GO TO 2100	00002790
IF (A(I,J).GT.0.0) Q=Q+A(I,J)	00002800
2100 CONTINUE	00002810
2110 IF (Q.LT.0.0) GO TO 3000	00002820
K=NF(I)	00002830
DO 2200 K1=1,K	00002840
J1=JF(I,K1)	00002850
IF (NS(J1).EQ.0) GO TO 2200	00002860
2120 IF (Q.GE.ABS(A(I,J1))) GO TO 2200	00002870
NS(J1)=0	00002880
L=L+1	00002890
SB(L)=BCIB	00002900
IF (A(I,J1).GT.0.0) GO TO 2150	00002910
S(L)=-J1	00002920
GO TO 2200	00002930
2150 S(L)=J1	00002940
ZS=ZS+C(J1)	00002950
DO 2175 I1=1,M	00002960
2175 BS(I1)=BS(I1)+A(I1,J1)	00002970
KINS=KINS+1	00002980
2200 CONTINUE	00002990
IF (KINS.GE.IINS) GO TO 1920	00003000

2220	CONTINUE	00003010
	IF (MSC.EQ.0) GO TO 2025	00003020
	IF (KINS.EQ.0) GO TO 1000	00003030
	IJK=2	00003040
	GO TO 1920	00003050
C 4A		00003060
2320	CONTINUE	00003070
	IF (M.EQ.M0) GO TO 2340	00003080
	DO 2325 I=M1,M	00003090
	B(I)=B(I)+ZS-ZKBAR-ZBAR	00003100
2325	BS(I)=BS(I)+ZS-ZKBAR-ZBAR	00003110
2340	ZBAR=ZS-ZKBAR	00003120
	DO 2350 J=1,N	00003130
2350	SMAX(J)=S(J)	00003140
	GO TO 3300	00003150
2400	K1=0	00003160
	DO 2500 J=1,N	00003170
	IF (NS(J).EQ.0) GO TO 2500	00003180
	IF (ITB.EQ.0) GO TO 2430	00003190
	IF (ZS+C(J).GE.ZBAR) GO TO 2500	00003200
	DO 2450 I=1,M	00003210
	IF (A(I,J).LE.0.0) GO TO 2450	00003220
	IF (BS(I).GE.0.0) GO TO 2450	00003230
2430	CONTINUE	00003240
	K1=K1+1	00003250
	T(K1)=J	00003260
	GO TO 2500	00003270
2450	CONTINUE	00003280
2500	CONTINUE	00003290
	IF (K1.EQ.0) GO TO 3200	00003300
	NAP=NAP+1	00003310
	P=-1.0E10	00003320
	DO 2575 K=1,K1	00003330
	J=T(K)	00003340
	P1=0.0	00003350
	DO 2550 I=1,M	00003360
	P2=BS(I)+A(I,J)	00003370
	IF (P2.GE.0.0) GO TO 2550	00003380
	P1=P1+P2	00003390
2550	CONTINUE	00003400
	IF (P1.LE.P) GO TO 2575	00003410
	P=P1	00003420
	J1=J	00003430
2575	CONTINUE	00003440
	NS(J1)=0	00003450
	L=L+1	00003460
	S(L)=J1	00003470
	ZS=ZS+C(J1)	00003480
	DO 2600 I=1,M	00003490
2600	BS(I)=BS(I)+A(I,J1)	00003500
	H(L)=H(L)+1.0	00003510
	GO TO 1910	00003520
3000	NCON=NCON+1	00003530
C	PRINT 3010,I	00003540
3010	FORMAT (1H0,I3,26H(TH) CONSTRAINT INFEASIBLE)	00003550
	GO TO 3500	00003560
3100	NRED=NRED+1	00003570
C	PRINT 3110	00003580
3110	FORMAT (33H0Z CANNOT BE REDUCED (ZS GE ZBAR))	00003590
	GO TO 3500	00003600

3200	NAUG=NAUG+1	00003610
C	PRINT 3210	00003620
3210	FORMAT (25HONO AUGMENTATION POSSIBLE)	00003630
	GO TO 3500	00003640
3300	NOPT=NOPT+1	00003650
	CALL DATIME (0,IT3)	00003660
	IF (NOP.NE.0) GO TO 3500	00003670
	PRINT 3310,ZS	00003680
	PRINT 3600,((S(K),SB(K)),K=1,L)	00003690
3310	FORMAT (23H0 BETTER SOLUTION FOUND,5X,2HZ=,1PE15.8)	00003700
	GO TO 3500	00003710
3400	NLPF=NLPF+1	00003720
	GO TO 3500	00003730
C 48		00003740
3500	CONTINUE	00003750
	NENUM=NENUM+1	00003760
	IF (NENUM.LT.KENUM) GO TO 3530	00003770
	NENUM=0	00003780
3505	CONTINUE	00003790
	ENUM=0.0	00003800
	DO 3510 K=1,N	00003810
3510	IF (SB(K).EQ.BCIB) ENUM=ENUM+.5**K	00003820
	CALL DATIME (0,IT2)	00003830
	ELT1=IT2-ITO	00003840
	ELT2=IT2-IT1	00003850
	IT1=IT2	00003860
	ELT1=ELT1/1000.0	00003870
	ELT2=ELT2/1000.0	00003880
	IF (IT2-ITO.LT.MAXT) GO TO 3515	00003890
	MAXT=-1	00003900
	GO TO 3517	00003910
3515	CONTINUE	00003920
	IF (NOP.NE.0) GO TO 3700	00003930
3517	CONTINUE	00003940
	PRINT 3520,ENUM,ELT1,ELT2,L	00003950
3520	FORMAT (11H0,F10.5,38H OF THE SOLUTIONS HAVE BEEN ENUMERATED,5X,	00003960
	* 15HTIME IN SECONDS,2X,5HTOTAL,F8.3,2X,7HELAPSED,F8.3,	00003970
	* 5X,2HL=,13)	00003980
3530	CONTINUE	00003990
	IF (MAXT.LT.0) PRINT 3600,((S(K),SB(K)),K=1,L)	00004000
3600	FORMAT (15(2X,14,A1))	00004010
	IF (MAXT.LT.0) GO TO 3738	00004020
C 48		00004030
3700	NFATH=NFATH+1	00004040
3710	IF (SB(L).EQ.BLANK) GO TO 3900	00004050
	J=IABS(S(L))	00004060
	NS(J)=J	00004070
	IF (S(L).LT.0) GO TO 3735	00004080
	ZS=ZS-C(J)	00004090
	DO 3725 I=1,M	00004100
3725	BS(I)=BS(I)-A(I,J)	00004110
3735	SB(L)=BLANK	00004120
	S(L)=0	00004130
	L=L-1	00004140
	IF (L.GT.0) GO TO 3710	00004150
C FINISHED		00004160
3738	CONTINUE	00004170
	PRINT 3739,H1,H2	00004180
3739	FORMAT (1H1,5X,2A6)	00004190
	DO 3740 J=1,N	00004200

3740 S(J)=0	00004210
DO 3742 J=1,N	00004220
K=IABS(SMAX(J))	00004230
IF (K.EQ.0) GO TO 3744	00004240
3742 S(K)=1	00004250
3744 DO 3746 K=1,N	00004260
IF (S(K).NE.0) GO TO 3746	00004270
SMAX(J)=-K	00004280
J=J+1	00004290
3746 CONTINUE	00004300
CALL DATIME (0,IT2)	00004310
ELT1=IT2-ITO	00004320
ELT1=ELT1/1000.0	00004330
IF (MAXT.LT.0) GO TO 3752	00004340
PRINT 3750,ELT1	00004350
3750 FORMAT (30H0IMPLICIT ENUMERATION COMPLETE,5X,11HTOTAL TIME=,F8.3)	00004360
GO TO 3758	00004370
3752 PRINT 3755,ELT1	00004380
3755 FORMAT (14H0TIME EXCEEDED,5X,11HTOTAL TIME=,F8.3)	00004390
3758 CONTINUE	00004400
ZBAR=ZBAR+ZKBAR	00004410
PRINT 3760,ZBAR	00004420
3760 FORMAT (32H0LEAST Z AFTER VARIABLE CHANGE =,1PE15.8)	00004430
I=0	00004440
3800 DO 3810 K=1,N	00004450
3810 T(K)=0	00004460
DO 3820 K=1,N	00004470
K1=IABS(SMAX(K))	00004480
3820 IF (SMAX(K).GT.0) T(K1)=K1	00004490
PRINT 3830,(T(K),K=1,N)	00004500
3830 FORMAT (15(4X,13))	00004510
IF (I.NE.0) GO TO 3845	00004520
ZBAR=0.0	00004530
DO 3835 J=1,N	00004540
K=IABS(SMAX(J))	00004550
IF (CS(K).LT.0.0) SMAX(J)=-SMAX(J)	00004560
IF (SMAX(J).GT.0) ZBAR=ZBAR+CS(K)	00004570
3835 CONTINUE	00004580
PRINT 3840,ZBAR	00004590
3840 FORMAT (33H0LEAST Z BEFORE VARIABLE CHANGE =,1PE15.8)	00004600
I=1	00004610
GO TO 3800	00004620
3845 CONTINUE	00004630
ELT3=IT3-ITO	00004640
ELT3=ELT3/1000.0	00004650
NITER=NFATH+NFATH-1	00004660
PRINT 3850,NOPT,NRED,NCON,NAUG,NAP,NID,NLPP,NSIMP,NITER,ELT3	00004670
3850 FORMAT (23H0NO. FEASIBLE SOLUTIONS,15/	00004680
* 11H ZS GE ZBAR,15,6H TIMES/	00004690
* 22H CONSTRAINT INFEASIBLE,15,6H TIMES/	00004700
* 24H AUGMENTATION IMPOSSIBLE,15,6H TIMES/	00004710
* 22H AUGMENTATION POSSIBLE,15,6H TIMES/	00004720
* 14H INTEGER DUALS,15,6H TIMES/	00004730
* 12H P FATHOMED,15,6H TIMES/	00004740
* 10H P CALLED,15,6H TIMES/	00004750
* 15H NO. ITERATIONS,15/	00004760
* 26H LAST FEASIBLE SOLUTION AT,F8.3,9H SECONDS)	00004770
GO TO 100	00004780
3900 SB(L)=BCIB	00004790
S(L)=-S(L)	00004800

```

J=IABS(S(L))
IF (S(L).GT.0) GO TO 3950
ZS=ZS-C(J)
DO 3925 I=1,M
3925 BS(I)=BS(I)-A(I,J)
GO TO 1910
3950 ZS=ZS+C(J)
DO 3975 I=1,M
3975 BS(I)=BS(I)+A(I,J)
GO TO 1910
END
$IBFTC SIMPLE
C AUTOMATIC SIMPLEX REDUNDANT EQUATIONS CAUSE INFEASIBILITY
SUBROUTINE SIMPLE(INFLAG,MX,NN,A,B,C,KO,KB,P,JH,X,Y,OBJ,E,NOP)
REAL B(1),C(1),P(1),X(1),Y(1),OBJ
REAL E(90,90)
INTEGER INFLAG,MX,NN,KO(6),KB(1),JH(1)
EQUIVALENCE (XX,LL)
C THE FOLLOWING DIMENSION SHOULD BE THE SAME HERE AS IT IS IN CALLER.
REAL A(50,90)
REAL AA,A1JT,BB,COST,DT,RCOST,TEXP,TPIV,TY,XOLD,XX,XY,YI,YMAX,EM
INTEGER I,IA,INVC,IR,ITER,J,JT,K,KBJ,LL,M,N,JT2
INTEGER NCUT, NUMVR,NVER,NUMPV
LOGICAL TRIG,VER
LOGICAL FINV,FFRZ,SCH
COMMON /BLS/MS(90),ZBAR
DIMENSION NF(90)
C SET INITIAL VALUES, SET CONSTANT VALUES
FINV = .FALSE.
TRIG = .FALSE.
ITER = 0
LPSEQ = LPSEQ+1
NUMVR = 0
NUMPV = 0
M = MX
N = NN
TEXP = .5**16
NVER = M/2 + 5
NCUT = 4*M + 10
IF (INFLAG.EQ.0) GO TO 1410
C IMPOSE CORRECT TEMPERATURE ON ROWS
FFRZ = .TRUE.
DO 1960 L=1,M
IF (MS(L).EQ.NF(L)) GO TO 1955
IF (MS(L)*NF(L).GT.0.OR.(MS(L).EQ.0.AND.X(L).GE.0.)) GO TO 1950
I=L
IF (NF(L).NE.0) GO TO 1925
1920 IF (JH(I).GT.0) GO TO 1930
C IF JH DISAGREES WITH MS DO SPECIAL PIVOT
IF (MS(L).GT.0.AND.JH(L).GE.(-M)) GO TO 1950
IF (MS(L).LT.0.AND.JH(L).LT.(-M)) GO TO 1950
C SPECIAL PIVOT, SWITCH SINGLETONS
1925 DO 1926 J=1,M
P(J) = P(J) + E(I,J)
E(I,J) = -E(I,J)
1926 CONTINUE
OBJ = OBJ + X(I)
X(I) = -X(I)
JHL = JH(L)
IF (JHL.GE.(-M)) JH(L) = -L-M
00004810
00004820
00004830
00004840
00004850
00004860
00004870
00004880
00004890
00004900
00004910
00004920
00004930
00004940
00004950
00004960
00004970
00004980
00004990
00005000
00005010
00005020
00005030
00005040
00005050
00005060
00005070
00005080
00005090
00005100
00005110
00005120
00005130
00005140
00005150
00005160
00005170
00005180
00005190
00005200
00005210
00005220
00005230
00005240
00005250
00005260
00005270
00005280
00005290
00005300
00005310
00005320
00005330
00005340
00005350
00005360
00005370
00005380
00005390
00005400

```

	IF (JH(L).LT.(-M)) JH(L) = -L	00005410
	GO TO 1950	00005420
C	DO FULL PIVOT ON SINGLETON	00005430
1930	JT = -I	00005440
	COST = P(I)	00005450
	IF (MS(I).GT.0) GO TO 1931	00005460
	JT = JT-M	00005470
	COST = 1.-COST	00005480
1931	EN = 1.	00005490
	GO TO 630	00005500
C	GET COLUMN(JT)	00005510
1932	SCH = .FALSE.	00005520
	IF (COST.GT.0.) GO TO 1938	00005530
1935	GO TO 1000	00005540
C	SELECT ROW(IR)	00005550
1936	IF (IR.NE.0.OR.SCH) GO TO 1940	00005560
	SCH = .TRUE.	00005570
1938	EN = -EN	00005580
	DO 1937 J=1,M	00005590
	Y(J) = -Y(J)	00005600
1937	CONTINUE	00005610
	GO TO 1935	00005620
1940	IF ((SCH.AND.ABS(COST).GT.TPIV).OR.IR.EQ.0) GO TO 1980	00005630
1941	IF (EN.GT.0.) GO TO 1945	00005640
	DO 1942 J = 1,M	00005650
	Y(J) = -Y(J)	00005660
1942	CONTINUE	00005670
1945	GO TO 901	00005680
C	PIVOT(IR,JT)	00005690
1950	NF(L) = MS(L)	00005700
1955	IF (JH(L).LT.0) GO TO 1960	00005710
	IA=JH(L)	00005720
	KB(IA)=L	00005730
1960	CONTINUE	00005740
	FFRZ = .FALSE.	00005750
	GO TO 910	00005760
C*	START WITH SINGLETON BASIS	00005770
1410	DO 1402 J=1,M	00005780
	KB(J) = 0	00005790
1402	CONTINUE	00005800
	FFRZ = .FALSE.	00005810
1400	DO 1401 I = 1,M	00005820
	JH(I) = -I	00005830
	NF(I) = MS(I)	00005840
	IF (NF(I).LT.0.OR.(NF(I).EQ.0.AND.B(I).LT.0.)) JH(I)=-I-M	00005850
1401	CONTINUE	00005860
C*	CREATE INVERSE FROM 'KB' AND 'JH' (STEP 7)	00005870
1320	VER = .TRUE.	00005880
	INVC = 0	00005890
	NUMVR = NUMVR +1	00005900
	TRIG = .FALSE.	00005910
	OBJ = 0.	00005920
	DO 1113 I = 1,M	00005930
	DO 1151 J=1,M	00005940
	E(J,I) = 0.	00005950
1151	CONTINUE	00005960
	IF (JH(I).LT.(-M)) GO TO 1111	00005970
	IF (JH(I).GT.0) JH(I) = 0	00005980
	E(I,I) = 1.	00005990
	P(I) = 0.	00006000

	X(I) = B(I)	00006010
	GO TO 1113	00006020
1111	E(I,I) = -1.	00006030
	P(I) = +1.	00006040
	OBJ = OBJ + B(I)	00006050
	X(I) = -B(I)	00006060
1113	CONTINUE	00006070
	DO 1102 JT = 1,N	00006080
	IF (KB(JT).EQ.0) GO TO 1102	00006090
	GO TO 600	00006100
C	GET COLUMN(JT)	00006110
1114	TY = TPIV	00006120
	IR = 0	00006130
	COST = C(JT)	00006140
	DO 1104 I = 1,M	00006150
	COST = COST + A(JT,I)*P(I)	00006160
	IF (JH(I).NE.0.OR.X(I).NE.0.OR.ABS(Y(I)).LE.TY) GO TO 1104	00006170
	TY = ABS(Y(I))	00006180
	IR = I	00006190
1104	CONTINUE	00006200
	IF (IR.NE.0) GO TO 1119	00006210
	TY = 0.	00006220
	DO 1105 I = 1,M	00006230
	IF (JH(I).NE.0.OR.X(I).EQ.0.OR.ABS(Y(I)).LE.TPIV) GO TO 1105	00006240
	IF (ABS(Y(I)).LE.TY*ABS(X(I))) GO TO 1105	00006250
	TY = ABS(Y(I)/X(I))	00006260
	IR = I	00006270
1105	CONTINUE	00006280
1119	IF (IR.NE.0) GO TO 900	00006290
C	PIVOT(IR,JT)	00006300
	FINV = .TRUE.	00006310
	IF (NOP.EQ.0) PRINT 1199,LPSEQ	00006320
1199	FORMAT(15HINVERT FAIL LP,14)	00006330
	GO TO 1410	00006340
1102	CONTINUE	00006350
C*	PERFORM A SIMPLEX ITERATION	00006360
1200	VER = .FALSE.	00006370
500	DO 503 I = 1,M	00006380
	IF (NF(I).EQ.0.AND.X(I).LT.0.) X(I)=0.	00006390
503	CONTINUE	00006400
C*	FIND MINIMUM REDUCED COST (STEP 3)	00006410
599	JT = 0	00006420
	BB = 0.0	00006430
	DO 701 J = 1,N	00006440
	IF (KB(J).NE.0) GO TO 701	00006450
	DT = C(J)	00006460
	DO 303 I = 1,M	00006470
	DT = DT + A(J,I)*P(I)	00006480
303	CONTINUE	00006490
	IF (DT.GE.BB) GO TO 701	00006500
	BB = DT	00006510
	JT = J	00006520
701	CONTINUE	00006530
	DO 702 I=1,M	00006540
	IF (JH(I).LT.0) GO TO 702	00006550
	IF (P(I).LT.BB) GO TO 703	00006560
	IF ((1.-P(I)).GE.BB) GO TO 702	00006570
	BB = 1.-P(I)	00006580
	JT = -I-M	00006590
	GO TO 702	00006600

703	BB=P(1)	00006610
	JT = -1	00006620
702	CONTINUE	00006630
	COST = BB	00006640
	IF (JT.EQ.0) GO TO 203	00006650
	IF (ITER.GE.NCUT) GO TO 160	00006660
	ITER = ITER +1	00006670
C*	MULTIPLY INVERSE TIMES A(.,JT)	(STEP 4)
	IF (JT.LT.0) GO TO 630	00006680
C	BEGIN SUBROUTINE GET COLUMN(JT)	00006690
600	DO 610 I = 1,M	00006700
	Y(I) = 0.0	00006710
610	CONTINUE	00006720
	DO 605 I = 1,M	00006730
	AIJT = A(JT,I)	00006740
	IF (AIJT.EQ.0.) GO TO 605	00006750
	DO 606 J = 1,M	00006760
	Y(J) = Y(J) + AIJT*E(J,I)	00006770
606	CONTINUE	00006780
605	CONTINUE	00006790
	GO TO 640	00006800
630	JT2 = -JT	00006810
	EM = 1.	00006820
	IF (JT2.LE.M) GO TO 631	00006830
	JT2 = JT2 - M	00006840
	EM = -1.	00006850
631	DO 632 I=1,M	00006860
	Y(I) = EM*E(I,JT2)	00006870
632	CONTINUE	00006880
640	YMAX = 0.	00006890
	DO 620 I = 1,M	00006900
	YMAX = AMAX1(ABS(Y(I)),YMAX)	00006910
620	CONTINUE	00006920
	TPIV = YMAX * TEXP	00006930
C	END OF GET COLUMN	00006940
	IF (FFRZ) GO TO 1932	00006950
	IF (VER) GO TO 1114	00006960
	RCOST = YMAX/BB	00006970
	IF (TRIG.AND.BB.GE.(-TPIV)) GO TO 203	00006980
	TRIG=BB.GE.(-TPIV)	00006990
C*	SELECT PIVOT ROW	(STEP 5)
1000	AA = TPIV	00007000
	IR = 0	00007010
1002	DO 1003 I = 1,M	00007020
	IF (X(I).NE.0..OR.Y(I).LE.AA..OR.NF(I).NE.0) GO TO 1003	00007030
	AA = Y(I)	00007040
	IR = I	00007050
1003	CONTINUE	00007060
	IF (IR.NE.0) GO TO 1020	00007070
	AA = 0.	00007080
	DO 1010 I = 1,M	00007090
	IF (NF(I).NE.0..OR.Y(I).LE.TPIV..OR.Y(I).LE.AA*X(I)) GO TO 1010	00007100
	AA = Y(I)/X(I)	00007110
	IR = I	00007120
1010	CONTINUE	00007130
1020	IF (FFRZ) GO TO 1936	00007140
	IF (IR.EQ.0) GO TO 207	00007150
C*	PIVOT ON (IR,JT)	(STEP 6)
901	IA = JH(IR)	00007160
	IF (IA.GT.0) KB(IA) = 0	00007170
		00007180
		00007190
		00007200

C	BEGIN SUBROUTINE PIVOT(IR,JT)	00007210
900	NUMPV = NUMPV + 1	00007220
	JH(IR) = JT	00007230
	IF (JT.GT.0) KB(JT) = IR	00007240
	YI = -Y(IR)	00007250
	Y(IR) = -1.0	00007260
	DO 904 J = 1,M	00007270
	XY = E(IR,J)/YI	00007280
	IF (XY.EQ.0.) GO TO 904	00007290
	P (J) = P (J) + COST * XY	00007300
	E(IR,J) = 0.	00007310
	DO 906 I = 1,M	00007320
	E(I,J) = E(I,J) + XY * Y(I)	00007330
906	CONTINUE	00007340
904	CONTINUE	00007350
	XY = X(IR) / YI	00007360
	DO 908 I = 1, M	00007370
	XOLD = X(I)	00007380
	X(I) = XOLD + XY * Y(I)	00007390
908	CONTINUE	00007400
	Y(IR) = -YI	00007410
	X(IR) = -XY	00007420
C	END OF PIVOT	00007430
	OBJ = OBJ + XY*COST	00007440
	IF (VER) GO TO 1102	00007450
C	EXCHANGE ROWS IF SLACK PIVOTED IN WRONG ROW	00007460
	IF (JT.GT.0.OR.JT2.EQ.IR) GO TO 907	00007470
	XY = X(IR)	00007480
	X(IR) = X(JT2)	00007490
	X(JT2) = XY	00007500
	DO 909 I = 1,M	00007510
	XY = E(IR,I)	00007520
	E(IR,I) = E(JT2,I)	00007530
	E(JT2,I) = XY	00007540
909	CONTINUE	00007550
	IA = JH(JT2)	00007560
	JH(JT2) = JT	00007570
	JH(IR) = IA	00007580
	KB(IA) = IR	00007590
907	INVC = INVC + 1	00007600
C	TO STEP 1 IF NOT INVERTING, TO STEP 7 IF INVERTING	00007610
	IF (FFRZ) GO TO 1950	00007620
	IF (OBJ.GE.ZBAR) GO TO 180	00007630
	IF (FINV) GO TO 1200	00007640
910	IF (INVC.GE.NVER) GO TO 1320	00007650
	GO TO 1200	00007660
C*	END OF ALGORITHM, SET EXIT VALUES ***	00007670
207	IF (RCOST.LE.(-1000.)) GO TO 203	00007680
C	INFINITE SOLUTION	00007690
	K = 2	00007700
	GO TO 250	00007710
180	K=6	00007720
	GO TO 250	00007730
C	PROBLEM IS CYCLING PERHAPS	00007740
160	K = 4	00007750
	PRINT 161,LPSEQ	00007760
161	FORMAT(31H0ITERATION LIMIT EXCEEDED ON LP,14)	00007770
	GO TO 250	00007780
C	FEASIBLE OR INFEASIBLE SOLUTION	00007790
203	K = 0	00007800

250 DO 1399 J = 1,M	00007810
XX = 0.0	00007820
KBJ = KB(J)	00007830
IF (KBJ.NE.0) XX = X(KBJ)	00007840
KB(J) = LL	00007850
1399 CONTINUE	00007860
KO(1) = K	00007870
KO(2) = ITER	00007880
KO(3) = INVC	00007890
KO(4) = NUMVR	00007900
KO(5) = NUPV	00007910
KO(6) = JT	00007920
IF (NOP.NE.0) RETURN	00007930
PRINT 162,LPSEQ,(KO(1),I=1,6)	00007940
162 FORMAT(3H LP,15,6H KO ,6I6)	00007950
C PRINT 1982	00007960
1982 FORMAT(21H01 JH NF MS ,P,Y,X,B/1X)	00007970
C DO 1983 I=1,M	00007980
C PRINT 1984,I,JH(I),NF(I),MS(I),P(I),Y(I),X(I),B(I)	00007990
1983 CONTINUE	00008000
1984 FORMAT(1X,4I3,4F12.6)	00008010
RETURN	00008020
1980 IF (NOP.EQ.0) PRINT 1981,LPSEQ,L,IR,SCH,COST	00008030
1981 FORMAT( 3H0LP,14,12H FAIL, SLACK,13,4H IR=13,5H SCH=L1,3H C=F19.6)	00008040
IF (IR.NE.0) GO TO 1941	00008050
GO TO 1410	00008060
END	00008070

REFERENCES

1. Geoffrion, A. M. (April 1967), "Integer Programming by Implicit Enumeration and Balas' Method," SIAM Review, 9, 2, pp. 178-190.
2. ----- (September 1967), "Implicit Enumeration Using An Imbedded Linear Program," RM-5406-PR, The RAND Corporation.
3. ----- (October 1967), "Recent Computational Experience With Three Classes of Integer Linear Programs," P-3699, The RAND Corporation.
4. Petersen, C. C. (May 1967), "Computational Experience With Variants of the Balas Algorithm Applied to the Selection of R&D Projects," Management Science, 13, 9, pp. 736-750.